-IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First-Named Inventor: Guido Frederiks	Docket No.: US030356US2
Application No.: 10/572,884 Conf.: 3000	Art Unit:
Date Filed: 03/20/2006 Customer No. **65913**	Examiner: To be determined

Title: METHOD AND APPARATUS FOR MASTER/SLAVE DIRECT MEMORY ACCESS HARDWARE AND SOFTWARE CONTROL

Commissioner for Patents
P.O. Box 1450
Arlington, VA 22313-1450

PETITION TO FILE APPLICATION WITH AN INVENTOR THAT CANNOT BE REACHED UNDER 37 CFR 1.47.

Sir:

In response to the Notice of Missing Requirements mailed on 06/25/2008. These efforts begun on August 7, 2008 were not successful. Attached is a Declaration presenting the facts of our effort.

Applicants believe that a sufficient effort has been made in attempting to obtain signature of the non-signing inventors. Therefore acceptance of present application with a non-signing inventor under 35 CFR 1.47(a) is earnestly requested.

Please charge any fees other than the issue fee and credit any overpayments to Deposit Account 50-4019.

Declaration of Facts Regarding Inventor's Unavailability begins on page 3.

Respectfully submitted,

Date: *18-DEC-2008*

By: */Aaron M Waxler/*
Aaron Waxler No. 48,027
(914) 860-4296

Correspondence Address:

Intellectual Property &
Licensing
NXP, B.V.
1109 McKay Drive; M/S-41SJ
San Jose, CA 95131 USA

CUSTOMER NO. 65913

DECLARATION OF FACTS REGARDING INVENTOR'S UNAVAILABILITY

I Vilimaina Naga declare that:

I am Aaron Waxler's IP Assistant responsible for preparing the Oath and Declaration for the above referenced patent application.

The present application is being prosecuted by NXP, B.V. previously, Philips Intellectual Property & Standards was managing this case as a part of the Koninklijke Philips Electronic, N.V. (KPENV). The NXP organization was spun-off from KPENV on October 1, 2006 as an independent entity.

Mr. Guido Frederiks is no longer employed by Koninklijke Philips Electronics, N.V. (KPENV) or NXP, B.V.

1)As described in Exhibit 1, August 7, 2008 I sent via FedEx Priority Overnight Service on behalf of Anthony Correa IP Assistant who is no longer with NXP the Oath and Declaration, and Assignment, Cover Letter, and a copy of the application having tracking number 790065589643, at 1623B Greenwood Ln, Monte Sereno, CA 95030 to his last known address that we have in our database.

2) As described in Exhibit 2, on August 8, 2008 FedEx delivered the package, and it was returned because no one by the name of Guido Frederiks resides at 1623B Greenwood Ln, Monte Sereno, CA 95030.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Respectfully submitted,

/Vilimaina Naga/

Vilimaina Naga
(408) 474-9067

EXHIBIT 1

From: Origin ID: RHVA (408)474-9067
vilimaina Naga
NXP Semiconductors
1109 McKay Drive M/S-41SJ

San Jose,, CA 95131

Ship Date: 07AUG08
ActWgt: 1 LB
System#: 1244941/INET8061
Account#: S *********

Delivery Address Bar Code

Ref # US030356US, US040131US
Invoice #
PO #
Dept # 9081

SHIP TO: 4084749060 BILL SENDER

Guido Frederiks

16230B GREENWOOD LN

MONTE SERENO, CA 950303053

TRK# 0201 7900 6558 9643

FRI - 08AUG A2
PRIORITY OVERNIGHT
RES

95030
CA-US
SJC

WA SJCA

After printing this label:
1. Use the 'Print' button on this page to print your label to your laser or inkjet printer.
2. Fold the printed page along the horizontal line.
3. Place label in shipping pouch and affix it to your shipment so that the barcode portion of the label can be read and scanned.

Warning: Use only the printed original label for shipping. Using a photocopy of this label for shipping purposes is fraudulent and could result in additional billing charges, along with the cancellation of your FedEx account number.

Use of this system constitutes your agreement to the service conditions in the current FedEx Service Guide, available on fedex.com.FedEx will not be responsible for any claim in excess of $100 per package, whether the result of loss, damage, delay, non-delivery,misdelivery,or misinformation, unless you declare a higher value, pay an additional charge, document your actual loss and file a timely claim.Limitations found in the current FedEx Service Guide apply. Your right to recover from FedEx for any loss, including intrinsic valueof the package, loss of sales, income interest, profit, attorney's fees, costs, and other forms of damage whether direct, incidental,consequential, or special is limited to the greater of $100 or the authorized declared value. Recovery cannot exceed actual documented loss.Maximum for items of extraordinary value is $500, e.g. jewelry, precious metals, negotiable instruments and other items listed in our ServiceGuide. Written claims must be filed within strict time limits, see current FedEx Service Guide.

EXHIBIT 1
CONT.

To GUIDO.FREDERIKS@PHILIPS.COM

cc

bcc

IP Department US/SVL/SC/PHILIPS
Sent by:
Anthony Correa

Subject URGENT: Signatures Required For USPTO Invention Submission

2008-08-04 11:12 AM

Docket No.: ~~US030350~~ Signatures Needed
Filing Date: 03/20/2006

Dear Sir:

Relative to the maintenance of this application here in the United States and with patent offices around the world, certain documents require your signature and date. I attached these documents for a patent application filed on 20MAR08. I also attached a copy of the final version of the patent application in PDF format. Please review all of these documents at your earliest convenience. Please verify these documents and advise me immediately if any error is found.

You will need to print and execute the file titled "Oath" and "Assign" using size A4 paper (certain patent offices require correspondence on this specific paper size). **Be sure to execute each page with your *signature* and *date*. Please do not print double sided.** Please do not sign and date on any other place other than where you have been asked to sign and date. Please return the executed documents to me at your earliest convenience (see below for return instructions). **If you do not have access to A4 paper, please still print and execute the documents and return them to me at your earliest convenience.**

Returning the executed documents:

Outside of San Jose:
Please use the Courier Address listed below and send the signed and dated documents to my attention using Federal Express (bill to account number 100488647).

In San Jose:
Please return the signed and dated documents to me using Interoffice mail at Mail Stop 41.

Also please send me a copy of the executed documents electronically via email or fax.

Please note: the attachment is an Adobe Acrobat file. Please do not sign and date these documents "*electronically*" as this is not honored as a valid signature by any patent office. Your actual signature and date is required.

Could you please send me the documents electronically via email before 10 AUG 2008, and than the hard copies in the mail at your earliest convenience.

Regards,

Anthony Correa
IP Assistant

IP Assistant

NXP Intellectual Property & Licensing
Visitor's Address: 1130 Ringwood Court, San Jose,CA 95131 USA
Mailing Address: 1109 McKay Drive MS41, San Jose, CA 95131 USA
Courier Address: 1140 Ringwood Court MS41, San Jose, CA 95131
Phone: +1 408 474 3000; Fax: +1 408 474 9082
E-Mail: ip.department.us@nxp.com

Intranet: pww.ips.nxp.com
Internet: www.ips.nxp.com

SOLE ASSIGNMENT

For good and valuable consideration, receipt of which is hereby acknowledged, I, as a below-named Assignor, hereby sell, assign, and transfer the entire and exclusive right, title, and interest in the following to **NXP B.V.**, having a place of business at High Tech Campus 60, NL-5656 AG, Eindhoven, NL, its successors, assigns, and legal representatives, including any nominees (collectively "the Assignee"):

(1) my invention relating to "**METHOD AND APPARATUS FOR MASTER/SLAVE DIRECT MEMORY ACCESS HARDWARE AND SOFTWARE CONTROL**" for which an US provisional patent application was filed in the United States Patent and Trademark Office on **09/23/2003,** having a Serial Number **60/505,069**.

(2) the foregoing application and all other United States, foreign and international patent applications associated therewith, based thereon, or claiming priority there from including, but not limited to, any and all provisional's, non-provisional's, divisions, continuations, continuations-in-part, re-examinations, reissues, and extensions thereof, and

(3) the right to claim priority thereto, and the entire and exclusive right, title, and interest in and to any and all patents granted on these applications.

I authorize and request that the Patent Office officials in the United States and in any and all foreign countries to issue any and all Letters Patent when granted, solely to **NXP B.V.**, for its sole use, and that of its successors, assigns, and legal representatives.

I will provide my cooperation to enable the Assignee to enjoy the foregoing right, title, and interest to the fullest extent. Upon request at the expense of the Assignee, I agree to execute all papers, take all rightful oaths, testify in all legal proceedings including patent prosecutional actions and infringement actions, and do all other such acts which may be necessary, desirable, or convenient for securing and maintaining patents on the foregoing invention or for perfecting title thereto in the Assignee.

I certify that I have the full right to convey the above rights.

_____ _____, Assignor

Date Guido FREDERIKS

Doc Code: OATH

PTO/SB/01 (05-0
Approved for use through 06/30/2010. OMB 0
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION (37 CFR 1.63)

☐ Declaration Submitted With Initial Filing OR ☑ Declaration Submitted after Initial Filing (surcharge (37 CFR 1.16 (f)) required)

Attorney Docket Number	US03 0356 US2
First Named Inventor	FREDERIKS, GUIDO
COMPLETE IF KNOWN	
Application Number	10/572,884
Filing Date	03/20/2006
Art Unit	
Examiner Name	

I hereby declare that: (1) Each inventor's residence, mailing address, and citizenship are as stated below next to their name and (2) I believe the inventor(s) named below to be the original and first inventor(s) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND APPARATUS FOR MASTER/SLAVE DIRECT MEMORY ACCESS HARDWARE AND SOFTWARE CONTROL

(Title of the Invention)

the application of which

☐ is attached hereto

OR

☑ was filed on (MM/DD/YYYY) 09/22/2004 as United States Application Number or PCT International Application Number PCT/IB2004/051819 and was amended on (MM/DD/YYYY) (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified application, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56, including for continuation-in-part applications, material information which became available between the filing date of the prior application and the national or PCT international filing date of the continuation-in-part application.

Authorization To Permit Access To Application by Participating Offices

☐ If checked, the undersigned hereby grants the USPTO authority to provide the European Patent Office (EPO), the Japan Patent Office (JPO), and any other intellectual property offices in which a foreign application claiming priority to the above-identified application is filed access to the above-identified patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, or other intellectual property office in which a foreign application claiming priority to the above-identified application is filed to have access to the application.

In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the application-as-filed with respect to: 1) the above-identified application, 2) any foreign application to which the above-identified application claims priority under 35 USC 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the above-identified US application, and 3) any U.S. application from which benefit is sought in the above-identified application.

In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date of filing the Authorization to Permit Access to Application by Participating Offices.

DECLARATION — Utility or Design Patent Application

Claim of Foreign Priority Benefits

I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or (f), or 365(b) of any foreign application(s) for patent, inventor's or plant breeder's rights certificate(s), or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent, inventor's or plant breeder's rights certificate(s), or any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached? YES	NO
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐

☐ Additional foreign application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

DECLARATION — Utility or Design Patent Application

Direct all correspondence to:	☑ The address associated with Customer Number:	65913	OR ☐	Correspondence address below

Name		
Address		
City	State	ZIP
Country	Telephone	Email

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available. Petitioner/applicant is advised that documents which form the record of a patent application (such as the PTO/SB/01) are placed into the Privacy Act system of records DEPARTMENT OF COMMERCE, COMMERCE-PAT-7, System name: *Patent Application Files*. Documents not retained in an application file (such as the PTO-2038) are placed into the Privacy Act system of COMMERCE/PAT-TM-10, System name: *Deposit Accounts and Electronic Funds Transfer Profiles*.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

NAME OF SOLE OR FIRST INVENTOR:	☐ A petition has been filed for this unsigned inventor		
Given Name (first and middle [if any]) GUIDO		Family Name or Surname FREDERIKS	
Inventor's Signature			Date
Residence: City MONTE SERENO	State CA	Country US	Citizenship NL
Mailing Address 16230B GREENWOOD LANE			
City MONTE SERENO	State CA	Zip 95030	Country US

☐ Additional inventors or a legal representative are being named on the _____ supplemental sheet(s) PTO/SB/02A or 02LR attached hereto

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 March 2005 (31.03.2005)

PCT

(10) International Publication Number
WO 2005/029341 A1

(51) International Patent Classification[7]: G06F 13/14

(21) International Application Number:
PCT/IB2004/051819

(22) International Filing Date:
22 September 2004 (22.09.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/505,069 23 September 2003 (23.09.2003) US

(71) Applicant *(for all designated States except US)*: KONINKLIJKE PHILIPS ELECTRONICS, N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(71) Applicant *(for AE only)*: U.S. PHILIPS CORPORATION [US/US]; 1251 Avenue of the Americas, New York, New York 10020 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: FREDERIKS, Guido [NL/US]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

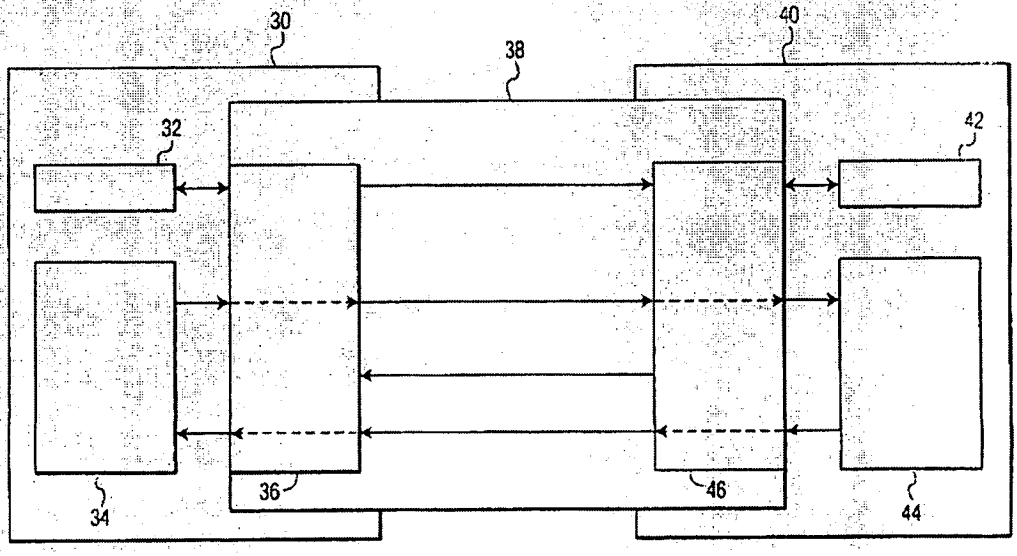
(74) Common Representative: KONINKLIJKE PHILIPS ELECTRONICS, N.V.; c/o LESTER, Shannon, 1109 McKay Drive, M/S-SJ41, San Jose, California 95131-1706 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI,

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR MASTER/SLAVE DIRECT MEMORY ACCESS HARDWARE AND SOFTWARE CONTROL



(57) Abstract: Systems and methods according to the present invention describe a robust DMA architecture including registers that flexibly support higher level hardware/software servicing needs during a DMA transfer. Among other things, such registers include mailbox registers and scratch pad registers which provide DMA architectures and methods according to the present invention with the capability to provide multiple interrupt channels and overflow data transfer capabilities that permit disparate hardware/software applications to reuse these DMA architectures with changes in their programming rather than changes in their architecture. Round robin DMA transfers using alternating memory locations are also described.

WO 2005/029341 A1

SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for the following designations AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW, ARIPO patent (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHODS AND APPARATUSES FOR MASTER/SLAVE DIRECT MEMORY ACCESS HARDWARE AND SOFTWARE CONTROL

The present invention relates generally to direct memory access (DMA) and, more particularly, to DMA hardware/software control techniques and structures that provide optimized data throughput.

Technologies associated with the communication of information have evolved rapidly over the last several decades. One of these communication technologies involves the transfer of data between memory devices. For example, the memory systems of various computer processing systems typically include a primary memory device and one or more larger, less expensive secondary memory devices. This permits large quantities of information to be stored in secondary memory for loading into the primary memory device when needed. Similar data transfers occur between memory devices in interconnected processing systems, which transfers are sometimes referred to as "master-slave" data transfers, and which are carried out over an interconnect device, e.g., a data bus. For example, the processor of a cellular telephone (master) may need to transfer data to and from subsystem processors, e.g., an antenna subsystem (slave).

One technique for performing such data transfers is known as direct memory access (DMA). DMA refers to a capability provided by some computer processing architectures that allows data to be sent directly from one memory device to another memory device without involving the main processor(s), thereby speeding up overall processing speeds. Usually a specified portion of memory is designated as an area to be used for DMA transfers. One problem with master-slave DMA data transfers occurs when a large number of data blocks need to be transferred between the master and slave devices. When this arises, conventional DMA data transfer techniques are challenged to perform such transfers efficiently and typically confront various latency issues. One example of such a latency issue is shown in the timeline of Figure 1. Therein, a first DMA data block is transferred from a master device to a slave device during the period DMA 1 and a second portion of the data block is transferred between the master and slave devices during the period DMA 2. Note that between time periods DMA 1 and DMA 2 there is a delay period denoted t1 in Figure 1. During this time, various handshaking activities occur between the master and slave devices. For example, at the end of DMA 1, the master device will signal a DMA controller indicating that it has transferred the first portion of DMA data. The DMA controller will then reset register(s) used in the DMA data transfer process, e.g., an address

register. After the handshaking is completed, the second DMA transfer (DMA2) can commence. This conventional DMA technique suffers from, among other things, an inefficiency in DMA transfer time due to the delay period t1.

One partial solution to this problem involves using two memory portions for DMA transactions an example of which is described in U.S. Patent No. 6,341,318, the disclosure of which is incorporated here by reference. Therein, with reference to Figures 2(a) and 2(b), in iteration 0, memory bank A is loaded (LD0) with a first portion of data from a data block. In iteration 1, a next portion of data is loaded (LD1) into memory bank B while the previously loaded portion in memory bank A is processed (P0) and then stored (ST0). In iteration 2, memory bank A is again loaded (LD2) with a third portion of data from a data block while the previously loaded portion in memory bank B is processed (P1) and then stored (ST1). The loading and processing/storing operations continue switching between each memory bank until all of the data block is loaded, processed, and stored. This technique may reduce latency associated with this type of DMA data transfer since the processing of portions of oversized DMA data blocks can be performed in parallel. However, the technique described in the '318 patent does not address the afore-described problem of handling a large number of DMA transfers between master and slave devices. Under such conditions, even using two DMA memory locations may be insufficient to eliminate latency associated with setting up each transfer. Moreover, the '318 patent provides relatively little control or status functionality in its register architecture. For example, no mechanisms are described for servicing higher level software protocols during the DMA transaction. As such, the DMA transaction architecture described in the '318 patent does not provide a DMA architecture that is sufficiently flexible to be reused without modifications between different hardware/software applications having different higher level servicing needs.

Accordingly, it would be desirable to provide more flexible techniques and devices for DMA data transfers which avoid the problems of conventional techniques.

Systems and methods according to the present invention address this need and others by providing a robust DMA architecture including registers that flexibly support higher level hardware/software servicing needs during a DMA transfer. Among other things, such registers include mailbox registers and scratch pad registers which provide DMA architectures and methods according to the present invention with the capability to provide multiple interrupt channels and overflow data transfer capabilities that permit disparate

hardware/software applications to reuse these DMA architectures with changes in their programming rather than changes in their architecture.

According to one exemplary embodiment of the present invention, a direct memory access (DMA) system for transferring information between a master device and a slave device comprises a first plurality of DMA channels for transferring information from the master device to the slave device; a second plurality of DMA channels for transferring information from the slave device to the master device; a first set of control registers which coordinate use of the first plurality of DMA channels to transfer information from the master device to the slave device; and a second set of control registers which coordinate use of the second plurality of DMA channels to transfer information from the slave device to the master device, wherein both the first set of control registers and the second set of control registers include at least one mailbox register for selectively providing an interrupt signal to a respective one of the master device and the slave device, in response to data written to the at least one mailbox register by the other of the master device and the slave device, during the DMA transaction.

According to another exemplary embodiment of the present invention, a method for direct memory access (DMA) information transfer between a master device and a slave device comprises the steps of: storing a first data portion received during a DMA transaction using a first DMA channel; storing a second data portion received during the DMA transaction using a second DMA channel; and selectively providing an interrupt signal to one of the master device and the slave device, by writing data in at least one mailbox register by the other of the master device and the slave device, during the DMA transaction.

The accompanying drawings illustrate exemplary embodiments of the present invention, wherein:

FIG. 1 depicts a timeline illustrating a DMA data transfer according to one conventional technique using a single memory area designated for DMA data transfers;

FIGS. 2(a) and 2(b) illustrate a conventional DMA data transfer using two memory areas;

FIG. 3 depicts a block diagram of a master-slave DMA architecture according to an exemplary embodiment of the present invention;

FIG. 4 shows an enlarged view of the slave portion of the DMA architecture of FIG. 3 including exemplary register groups;

FIG. 5 is a table providing an overview of various registers which can be included in DMA architectures according to exemplary embodiments of the present invention;

FIGS. 6(a)-6(l) illustrate exemplary individual register structures according to exemplary embodiments of the present invention;

FIG. 7 is a flowchart illustrating an exemplary method for transferring data from a master device to a slave device from the master's perspective according to an exemplary embodiment of the present invention;

FIG. 8 is a flowchart illustrating an exemplary method for transferring data from a master device to a slave device from the slave's perspective according to an exemplary embodiment of the present invention;

FIG. 9 is a flowchart illustrating an exemplary method for transferring data from a slave device to a master device from the slave's perspective according to an exemplary embodiment of the present invention; and

FIG. 10 is a flowchart illustrating an exemplary method for transferring data from a slave device to a master device from the master's perspective according to an exemplary embodiment of the present invention

The following detailed description of the invention refers to the accompanying drawings. The same reference numbers in different drawings identify the same or similar elements. Also, the following detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims.

In order to provide some context for this discussion, an exemplary master-slave system in which techniques and devices according to exemplary embodiments of the present invention can be implemented is depicted in the block diagram of Figure 3. Therein, the master device 30 includes a processor 32, memory device 34 and master DMA controller 36 (master DMA controller 36 may be logically viewed as part of the hardware connect layer 38). Similarly, the slave device 40 includes its own processor 42, memory device 44 and slave DMA controller 46 (slave DMA controller 46 may also be logically viewed as part of the hardware connect layer 38). The hardware connect layer 38 can be implemented as, for example, any suitable data bus that can convey the depicted functionality, e.g., a master to slave interrupt signal, a master to slave DMA data transfer signal, a slave to master interrupt signal and a slave to master DMA data transfer signal. Devices 34 and 44 are not limited to implementation as memory devices *per* se and can, alternatively, be any other devices which use data transferred via a bus from another device.

Figure 4 illustrates the slave DMA controller 46 in more detail and, in particular, shows a general grouping of various registers which can be used in accordance with exemplary embodiments of the present invention to coordinate DMA data transfers. A first (optional) group 48 of registers includes hardware connection layer 38 specific registers used by the master device 30 to control slave DMA and/or slave interrupt generation activity. For example, if an SDIO bus is used as hardware connect layer 38, the first group of registers 48 may include a register which indicates whether one bit mode or four bit mode is being used. Other examples include setting hardware specific registers to indicate whether DMA data transfers are occurring in full duplex mode or half duplex mode and/or the operating frequency. Typically, the hardware specific registers 48 (if any) will be only programmed during an initialization phase. A second group of registers 50 include master to slave mailbox and scratch registers which are used by the master device 30 to transfer software specific information to the slave device 40. As mentioned earlier, mailbox and scratch registers provide, among other features flexibility for higher level software and hardware protocols to operate during DMA activities. For example, when the master device 40 writes in a mailbox register, an interrupt signal can be sent to the slave processor 42. The generation of the interrupt signal responsive to writing in the mailbox register is programmable, as described below with respect to the control registers. The slave device 30 will then read the information stored in the mailbox register, which results in the interrupt being cleared. The information written to mailbox registers according to exemplary embodiments of the present invention may vary widely depending upon the master and slave devices being interconnected by the DMA architectures. For example, if the master device is a cellular telephone's central processor and the slave device is an antenna subsystem, the master device could write instructions to the antenna subsystem into the mailbox register(s), e.g., requesting the antenna subsystem to initiate low power consumption operation. Thus, the mailbox registers can be used to provide built-in interrupt channels which can be used in various ways depending upon the interconnected devices. The scratch registers are similar to the mailbox registers except that writing to a scratch register does not result in the generation of an interrupt signal. Thus, for example, the scratch registers can be used to carry overflow data associated with a command transferred between the master and slave devices via the mailbox register. Alternatively, the slave device 40 can find master device status information in a scratch register.

Register group 52 depicts mailbox and scratch registers for the slave to master direction. This group of registers performs similar functions to those described above with respect to group 50. When the slave device 40 writes in one of these mailbox registers, an interrupt signal can be sent to master device 30, subject to control register programming. When the master device 30 reads the corresponding mailbox, the interrupt will be cleared. Register group 54 depicts the DMA status/control registers which, among other things, control the amount of data that is transferred in one DMA cycle. In the master-to-slave direction, the control registers also indicate the address to which the data will be written, while in the slave-to-master direction the control registers also indicate the address from which the data will be read. As will be described in more detail below regarding individual registers and their functions, the status registers provide an indication of DMA activity as well as which interrupts are pending.

To better understand architectures according to exemplary embodiments of the present invention, a more detailed example of the various registers mentioned above will now be provided with respect to Figures 5-6(l). Those skilled in the art will appreciate that the details associated with this example are purely illustrative in nature and can be varied. Among other things, the addresses and bit lengths are purely exemplary. Figure 5 provides a table listing the programmable registers on the slave side of the hardware connect layer 38 including their type. Each of the registers listed in the table of Figure 5 is depicted in more detail in a corresponding one of Figures 6(a)-6(l).

Starting with Figure 6(a), an example of the master to slave interrupt status/control register is provided. This register type is also referred to herein as a "mailbox interrupt status/control register". This control/status register is used to control, and provide information regarding, the master-to-slave mailbox registers. Specifically, the slave device 40 can set a control bit associated with each of the master-to-slave mailbox registers to permit or deny interrupts to be generated when the corresponding mailbox register is written to by the master device 30. This feature can be used, for example, during initialization when disabling the mailbox interrupts may be desirable. After initialization, the slave device 40 may then enable the mailbox register interrupts. Status bits for each master-to-slave mailbox are also provided in this register, which permit the slave device 40 to identify which mailbox or mailboxes caused it to receive an interrupt. Once read, these status bits are reset. Similarly, Figure 6(b) depicts an interrupt status/control register for the slave-to-master direction. Thus, the master device 30 can set a control bit associated with each of

the slave- to-master mailbox registers to permit or deny interrupts to be generated when the corresponding mailbox register is written to by the slave device 30. Status bits for each slave-to-master mailbox are also provided in this register, which permit the master device 30 to check to see if an interrupt is pending.

Figure 6(c) depicts an exemplary DMA status/control register. This register can include, among other things, bits to store version information, bits indicating DMA activity for each DMA memory location in each direction, and bits for controlling which of the two DMA memory locations will be used for a next DMA transaction. With respect to this latter function, it can be seen in Figure 6(c) that the values for bits 0 and 1 in this register can be automatically inverted after each master-to-slave and slave-to-master transaction, respectively, to provide a round robin usage of the dual memory locations as described in more detail below. Figures 6(d)-6(g) provide examples of a master-to-slave mailbox register, a master-to-slave scratch register, a slave-to-master mailbox register and a slave-to-master scratch register respectively. Although multiple mailbox and scratch registers can be provided in each direction for DMA architectures according to the present invention, only one of each type/direction is described here since each of these types of registers can have the same structure and functionality. The number of bits allocated for storage of mailbox code and/or scratch values can be more or fewer than the 10 bits indicated in the examples shown in Figures 6(d)-6(g).

Figure 6(h) illustrates an exemplary DMA interrupt status/control register. This register provides for programmable enablement/disablement of the interrupt signals which are used to signal the completion of a DMA transfer. One bit is provided for each DMA memory location for each direction, for a total of four control bits in this exemplary embodiment of the present invention. Similarly, four bits provide status information for each DMA memory location for each direction regarding whether or not an interrupt has been set. The DMA interrupt status/control register can also be used to provide other functionality, for example providing an indicator regarding whether 16 bit data values are transmitted with their most significant bit (MSB) first or least significant bit (LSB) first. Figures 6(i)-6(l) illustrate exemplary master-to-slave DMA address register, master-to-slave DMA size register, slave-to-master DMA address register and slave-to-master DMA size register, respectively. One address and one size register can be provided for each DMA memory location used to transfer DMA data to specify the address and size of each memory

location. According to this exemplary embodiment, therefore, eight of these registers can be used as seen in Figure 5.

Having described a register architecture according to exemplary embodiments of the present invention, exemplary methods for transferring data using such an architecture will now be discussed with reference to the flowcharts of Figures 7-10, wherein Figures 7 and 8 depict a master-to-slave DMA transfer method and Figures 9 and 10 depict a slave-to-master DMA transfer method. On the master device's side, at step 70, the master device 30 first waits for the slave device 40 to initialize. This can be accomplished by, for example, the master device 30 polling one of the slave-to-master scratch registers and waiting for the slave device 40 to have written a predetermined value in that register. Next, at step 72, the master device 30 initializes the interrupts related to the master-to-slave mailboxes (Fig 6a). The slave device 40 uses these interrupts to indicate that a master-to-slave DMA channel has been initialized and is ready to receive data. The master device 30 then waits for data to send to the slave device 30 at step 74. At this step, the master device 30 knows that the slave device 40 has two master-to-slave DMA channels available. When master device 30 has data that needs to be transferred to the slave device 40, it then performs a DMA transfer at step 76. In a typical implementation of the slave DMA controller 46, the master device 30 does not have to keep track of which DMA channel the DMA data needs to be sent. The slave DMA controller 46 will automatically direct the incoming master-to-slave DMA data to the appropriate, next DMA channel. For this purpose the slave DMA controller 46 can have some hardware connect layer specific registers which are accessible by the master device 30 to enable this functionality. At step 78, the master device 30 is waiting for more data to send to the slave device 30 or for reception of an interrupt indicating that the slave device 30 has freed up the previously used DMA channel. Depending on which of the two indications comes in first, the master device 30 will change to the appropriate state, i.e., return to step 74 wherein two DMA channels are available or move on to step 80. Typically a slave-to-master mailbox interrupt is used by the slave device 40 to indicate to the master device 30 that a channel has been freed up again. However, there are other mechanisms available as well. For example, the master device 30 can periodically poll a scratch register, where the slave programmed value can be used to derive if the slave device 40 has any master-to-slave DMA channels available. Assuming that the flow next moves to step 80, the master device 30 once again has data that needs to be transferred to the slave device 40, and it sends the data via the DMA channel selected by the slave DMA controller 46.

The flow then proceeds to step 82, wherein the master device 30 knows that the slave device has no free master-to-slave DMA channels available. The master device 30 then waits for an indication (interrupt or polling) from the slave that a DMA channel is freed up again and the flow returns to step 78.

Figure 8 shows an exemplary process which occurs during a master-to-slave DMA transfer from the slave device 40's perspective. Therein, the slave device 40 initializes the master-to-slave interface at step 84. This step might, for example, include programming some hardware connection layer specific registers 48. The generation of master-to-slave DMA interrupts (Fig. 6(h)) will also typically be enabled at step 84, as well as setting the locations for the two master-to-slave DMA channels in the respective registers. When all of the initialization is done, the slave device 40 indicates to the master device 30 that the initialization has finished by, for example, writing a predetermined value to one of the scratch registers. Next, at step 86, the slave device 30 waits for the master device's DMA data. In this state the slave has two master-to-slave DMA channels available. When the slave device 40 receives a master-to-slave DMA end interrupt the slave device 40 can determine from the DMA status register (Fig. 6(h)) for which master-to-slave DMA channel the interrupt was generated. With that information the slave can find the memory location of the DMA data and the size from the channel corresponding size and address registers (Figs. 6(i) and 6(j)). The slave software, e.g., antenna subsystem software, can then be triggered to process the received data.

The process then moves to step 88. Therein, the slave device 40 can check to determine if it has a new memory location available to reprogram the memory address (Fig. 6(i))) for the DMA channel which has previously received DMA data, e.g., if that DMA channel has not been cleared yet. If so, the address register for this DMA channel is updated, and a signal is sent to the master device 30 to indicate that a new channel is available. This can be accomplished by, for example, generating a slave-to-master mailbox interrupt to indicate to the master that a DMA channel has been freed up again. As mentioned above, however, polling or hybrid polling/interrupt based schemes can be used for this purpose as well. If no memory location was available and a new DMA end interrupt is received, the slave device 30 can derive from the DMA status register (Fig. 6(h)) for which master-to-slave DMA channel the interrupt was generated. With that information the slave device 30 can find the memory location of the DMA data and the size from the

corresponding size and address registers (Figs. 6(i) and 6(j)). The slave software that will process the just received data will then be triggered.

The slave device 30 then moves to step 90 wherein it again waits for new memory to become available in which master-to-slave DMA package could be stored. When this memory becomes available, the address register for the next master to slave DMA channel is updated, and a trigger is generated to the master to indicate that a new channel is available. Those skilled in the art will appreciate that, instead of only a round robin approach using two memory locations for DMA transfers, the present invention can use three or more memory locations to implement the two DMA channels described above. For example, at steps 88 and 90 memory addresses which are different than those used at step 84 can be programmed into one or both of the address registers. Using more than two memory locations for master-slave DMA transactions can further reduce the latency, particularly for DMA transactions involving numerous data packages.

Figure 9 depicts a slave-to-master DMA data transfer method according to an exemplary embodiment of the present invention from the slave device 40's point of view. Therein, at step 92, the slave device 30 initializes the master-to-slave interface. This step can, for example, include programming some hardware connection layer specific registers 48. The master-to-slave DMA interrupts (Fig. 6(h)) can be enabled depending on later described communication algorithms. When the initialization is completed, the slave device 40 needs to indicate this to the master device 30. This can be accomplished by writing a predetermined value to one of the scratch registers. At step 94, the slave device 40 is waiting for an indication that the master device 40 has been initialized. This can be accomplished by, for example, the master device 30 polling one of the master-to-slave scratch registers and waiting for the slave to write a predetermined value in that register. Next, at step 96, the slave device 40 waits for data to send to the master device 30 and has two slave-to-master DMA channels available. When that data becomes available, at step 98, the slave device 40 initializes the address and size registers (Figs. 6(k), 6(l)) for the next DMA channel (Fig. 6(c)), and generates an indication to the master that a next slave-to-master DMA transfer has been setup. This indication can be an interrupt that is generated by writing to a mailbox register or updating a value in a scratch registers that the master polls, from which the master device can derive that new DMA data transfer has been setup. Upon reception of this indication, the master will initiate the master-to-slave DMA transfer. As mentioned earlier, the slave DMA controller 46 can automatically retrieve the slave-to-

master DMA data from the valid channel without any intervention from the master device 30.

At step 100, the slave device 40 is waiting for more data to send to the master or for the reception of an indication that the master device 30 has retrieved the slave-to-master DMA data of the initialized channel. This indication can be provided by using the TX DMA end interrupt in the register of Fig. 6(h). This interrupt will indicate which channel data has been retrieved. Another method for providing this indication is that after retrieving the DMA data, the master device 30 writes to a master-to-slave mailbox register, which will generate an interrupt for the slave device 40. A third method is that the slave device 40 polls a master-to-slave scratch register, whose value will be updated by the master device 30 after retrieving the DMA data. Assuming that the slave has more data for the master prior to receiving an indication that the previous slave-to-master DMA transfer has been completed, the process transitions to step 102. Therein, the slave device 40 initializes the address and size registers (Figs. 6(k) and 6(l)) for the still available DMA channel, and generates a previously described indication to the master device 30 that a next slave-to-master DMA transfer has been setup. The flow then moves to step 104, wherein the slave device 40 has no free slave-to-master DMA channels available and, therefore, waits for an indication (interrupt or polling) from the master device 30 that a DMA channel is freed up again.

Turning now to the master device's perspective of the slave-to-master DMA transfer in Figure 9, the master device 30 waits for the slave device 40 to initialize at step 106. This can be accomplished by, for example, the master device 30 polling one of the slave-to-master scratch registers and waiting for the slave device 40 to write a predetermined value in that register. Next, the master device 30 initializes the slave DMA controller 46 at step 108. This step can, for example, include programming hardware connection layer specific registers 48. The generation of slave-to-master DMA interrupts (Fig. 6(h)) will also be enabled (if needed). When the initialization is completed, the master device 30 indicates this to the slave device 40. This indication can, for example, be accomplished by writing a predetermined value to one of the master-to-slave scratch registers.

The master device 30 then waits for a slave-to-master DMA data indication at step 110. In this state there are no slave-to-master channels pending with data to be read. There are several ways that the data indication can be generated. For example, the slave device 40 can indicate that it has data ready to transfer via an interrupt that is generated by the slave writing to a slave-to-master mailbox register. Alternatively, the slave device 40 can update

a value in a scratch register that the master device 30 polls, from which the master can determine that new DMA data has been setup. Upon reception of this indication, the master device 30 will initiate the master-to-slave DMA transfer. As mentioned earlier, according to exemplary embodiments of the present invention, the master device 30 does not have to keep track of the DMA channel from which the DMA data needs to be retrieved. Instead, the slave DMA controller 46 can automatically retrieve the slave-to-master DMA from the valid channel. At step 112, the slave-to-master DMA transfer is ongoing. The master device 30 is waiting for either the end of DMA transfer indication (which is generated internally in the master), or a new indication from the slave device 40 that yet another DMA channel has been setup. When the end of DMA transfer indication is received, the master device 30 can trigger the function that will process the retrieved data. Depending on which 'end of slave to master DMA transfer' indication the slave uses, the master can write to a master-to-slave mailbox or scratch register to indicate that this transfer finished. When a new slave-to-master DMA data indication is received by the master device 30, and the previous DMA transfer has not yet finished, the process moves to step 114. Therein, the slave-to-master DMA is ongoing. The master is waiting for the end of DMA transfer indication (which is generated internally in the master). When the end of DMA transfer indication is received, the master can trigger the function that will process the retrieved data. Again, depending on which 'end of slave to master DMA transfer' indication the slave uses, the master device 30 can write to a master-to-slave mailbox or scratch register to indicate that this transfer finished, which returns the flow to step 100.

In the exemplary DMA methods described above, it is assumed that the slave DMA controller 46 has four parallel DMA channels, but the master device 30 does not. If the master device 30 also has four parallel DMA channels, the master device's flow diagrams would be similar to those of the slave device 40. Additionally, providing four parallel DMA channels for the master device would enable the master device 30 to setup several DMA transfers in parallel. Although the foregoing examples depict two DMA channels in each direction, those skilled in the art will appreciate that DMA architectures according to the present invention are not so limited. Instead, embodiments of the present invention could contain four, eight, sixteen or more DMA channels in each direction with a corresponding increase in the number of registers used to implement the architecture.

According to exemplary embodiments, the master device 30 initiates slave-to-master DMA transfers. If the slave device 40 has not initialized any slave-to-master DMA

channels with a new memory address and size, the slave DMA controller 46 will automatically respond to the slave-to-master DMA transfer request with the indication that the DMA size is zero. As mentioned above, initiating a slave-to-master DMA data transfer can be accomplished in at least three different ways. First, a polling based initialization can be used. If this technique is used, then the master device 30 will periodically start a slave-to-master DMA command sequence and will read, among other things, the slave-to-master size registers to determine if any DMA data has been stored for transfer. If data is available, then the process outlined above is performed, otherwise the master device 30 awaits the next polling interval.

Alternatively, an interrupt based initialization can be used. After the slave device 40 has programmed the slave-to-master address and size registers, a mailbox register is then written to in order to generate an interrupt to the master device 30. This provides yet another example of the flexibility created by providing mailbox registers to DMA architectures according to the present invention. On reception of the interrupt generated by the mailbox register, the master will initiate a slave-to-master DMA sequence. Different mailbox registers can be used to initiate transfers from two different DMA memory locations. The slave device 40 should first check the interrupt status bits in the mailbox interrupt status/control register to ensure that an interrupt is not already pending for the mailbox registers it intends to use to signal the master device 30 to initiate the slave-to-master DMA data transfer.

Thirdly, a hybrid polling/interrupt based initialization technique can be used for slave-to-master DMA transfers according to exemplary embodiments of the present invention. To begin, the slave device 40 again writes to a mailbox register to generate an interrupt to the master 30 which is interpreted as a request to initiate a slave-to-master DMA transfer. While the transfer is occurring from the first memory location, the slave device 40 programs the registers and second memory location with the next DMA package to be transferred. However, instead of using another mailbox register to send an interrupt signal to the master device 30 to start the transfer of the second data portion, the master device 30 automatically initiates the second slave-to-master transfer sequence. The slave device 30 then uses the interrupt it receives upon completion of the first transfer to begin preparing the third data portion for transfer. This process continues until such time as the slave device 40 runs out of data associated with this DMA transaction at which time it sets the size register

to zero. When the master device 30 requests the next block of data it will receive the size zero indication and end the DMA transaction.

Those skilled in the art will appreciate that the foregoing DMA architectures and methods provide sufficient programmability for interrupts and data transfers that they lend themselves to reuse without architectural changes between different types of master devices, slave devices, hardware interconnects and software applications. The ability to program the mailbox registers to selectively enable or disable interrupt generation, trigger the reading of scratch registers, etc., allows software developers to reuse code associated with this DMA architecture since it can maintain its register structure across different implementations.

The above-described exemplary embodiments are intended to be illustrative in all respects, rather than restrictive, of the present invention. Thus the present invention is capable of many variations in detailed implementation that can be derived from the description contained herein by a person skilled in the art. All such variations and modifications are considered to be within the scope and spirit of the present invention as defined by the following claims. No element, act, or instruction used in the description of the present application should be construed as critical or essential to the invention unless explicitly described as such. Also, as used herein, the article "a" is intended to include one or more items.

CLAIMS

What is claimed is:

1. A direct memory access (DMA) system for transferring information between a master device (30) and a slave device (40) comprising: a first plurality of DMA channels for transferring information from said master device (30) to said slave device (40); a second plurality of DMA channels for transferring information from said slave device (40) to said master device (30); a first set of control registers (Figs. 6a, 6c, 6d, 6e, 6h, 6i, 6j) which coordinate use of said first plurality of DMA channels to transfer information from said master device (30) to said slave device (40); and a second set of control registers (Figs. 6b, 6c, 6f, 6g, 6h, 6k, 6l) which coordinate use of said second plurality of DMA channels to transfer information from said slave device (40) to said master device (30), wherein both said first set of control registers and said second set of control registers include at least one mailbox register (Figs. 6d, 6f) for selectively providing an interrupt signal to a respective one of said master device (30) and said slave device (40), in response to data written to said at least one mailbox register by the other of said master device and said slave device, during said DMA transaction.

2. The system of claim 1, wherein each of said first and second sets of control registers further comprise a mailbox interrupt status/control register (Figs. 6a,6b) for enabling/disabling mailbox interrupt generation for each of said at least one mailbox registers and for indicating whether one of said at least one mailbox interrupts is pending for each of said at least one mailbox registers.

3. The system of claim 1, wherein each of said first and second sets of control registers further comprise at least one scratch register (Figs. 6e,6g) for storing data which can be read in response to said interrupt signal generated by said at least one mailbox register.

4. The system of claim 1, further comprising a DMA interrupt status/control register (Fig. 6h) for enabling/disabling data transfer complete interrupt generation associated with a completion of transferring said information and for indicating whether one of said data transfer complete interrupts is pending for each of said first and second DMA channels.

5. The system of claim 1 further comprising: a first address register for storing a first address associated with one of said first plurality of DMA channels; a second address register for storing a second address associated with another of said first plurality of DMA

channels; a third address register for storing a third address associated with one of said second plurality of DMA channels; and a fourth address register for storing a fourth address associated with another of said second plurality of DMA channels.

6. The system of claim 1, further comprising: a first size register for storing a size of said information transferred using one of said first plurality of DMA channels; a second size register for storing a size of said information transferred using another of said first plurality of DMA channels; a third size register for storing a size of said information transferred using one of said second plurality of DMA channels; and a fourth size register for storing a size of said information transferred using another of said first plurality of DMA channels.

7. The system of claim 1, further comprising: a DMA status/control register (Fig. 6c) for indicating, for each of said first and second plurality of DMA channels, whether DMA transfer activity is ongoing.

8. The system of claim 1, wherein said data written to said at least one mailbox register is a command requesting that more data be transferred.

9. The system of claim 1, wherein said data written to said at least one mailbox register is a command to activate a feature of one of said slave device and said master device.

10. The system of claim 9, wherein said feature is a reduce power consumption feature.

11. A method for direct memory access (DMA) information transfer between a master device (30) and a slave device (40) comprising the steps of: storing a first data portion received during a DMA transaction using a first DMA channel; storing a second data portion received during said DMA transaction using a second DMA channel; and selectively providing an interrupt signal to one of said master device (30) and said slave device (40), by writing data in at least one mailbox register (Figs. 6d, 6f) by the other of said master device (30) and said slave device(40), during said DMA transaction.

12. The method of claim 11, further comprising the step of: enabling/disabling mailbox interrupt generation for each of said at least one mailbox registers (Figs. 6d,6f) and for indicating whether one of said at least one mailbox interrupts is pending for each of said at least one mailbox registers using a mailbox interrupt status/control register (Figs. 6a, 6b).

13. The method system of claim 11, further comprising the step of: reading data stored in at least one scratch register (Figs. 6e, 6g) in response to said interrupt signal.

14. The method of claim 11, further comprising the step of: enabling/disabling data transfer complete interrupt generation associated with a completion of storing said first and second data portions and for indicating whether one of said data transfer complete interrupts is pending for each of said first and second DMA channels using a DMA interrupt status/control register (Fig. 6h).

15. The method of claim 11 further comprising the steps of: storing a first address associated with said first DMA channel; and storing a second address associated with said second DMA channel.

16. The method of claim 11, further comprising the steps of: storing a size of said first data portion; and storing a size of said second data portion.

17. The method of claim 11, further comprising the step of: providing an indication regarding whether transfer activity is currently ongoing in said first and second DMA channels.

18. The method of claim 11, wherein said data written to said at least one mailbox register is a command requesting that more data be transferred.

19. The method of claim 11, wherein said data written to said at least one mailbox register is a command to activate a feature of one of said slave device and said master device.

20. The method of claim 19, wherein said feature is a reduce power consumption feature.

21. The method of claim 15, further comprising the step of: storing a third data portion using said first DMA channel.

22. The method of claim 21, further comprising the step of: reprogramming a register which contains said first address with a third address that is different than said first address prior to said step of storing said third data portion.

23. The method of claim 21, further comprising the step of: storing a fourth data portion using said second DMA channel.

24. The method of claim 23, further comprising the step of: reprogramming a register which contains said second address with a fourth address that is different than said second address prior to said step of storing said fourth data portion.

25. The system of claim 1, wherein each of said first and second sets of control registers further comprise at least one scratch register for storing status information which can be read by one of said master device and said slave device

PCT/IB2004/051819

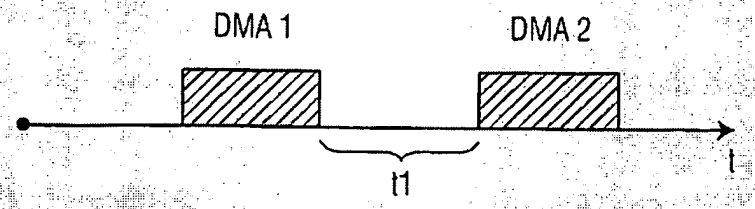


FIG. 1
PRIOR ART

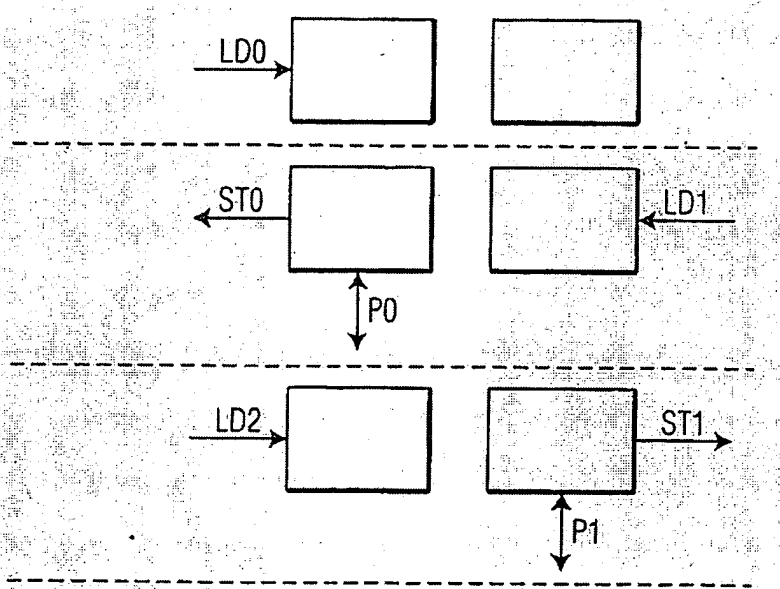


FIG. 2A
PRIOR ART

PCT/IB2004/051819

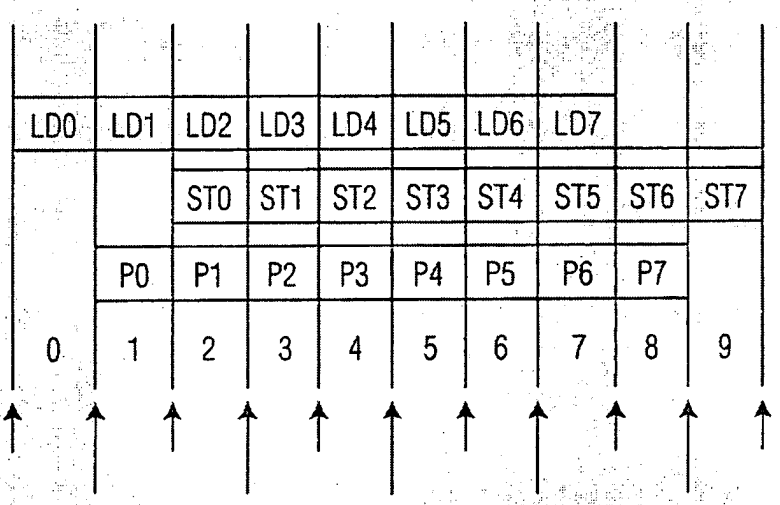


FIG. 2B
PRIOR ART

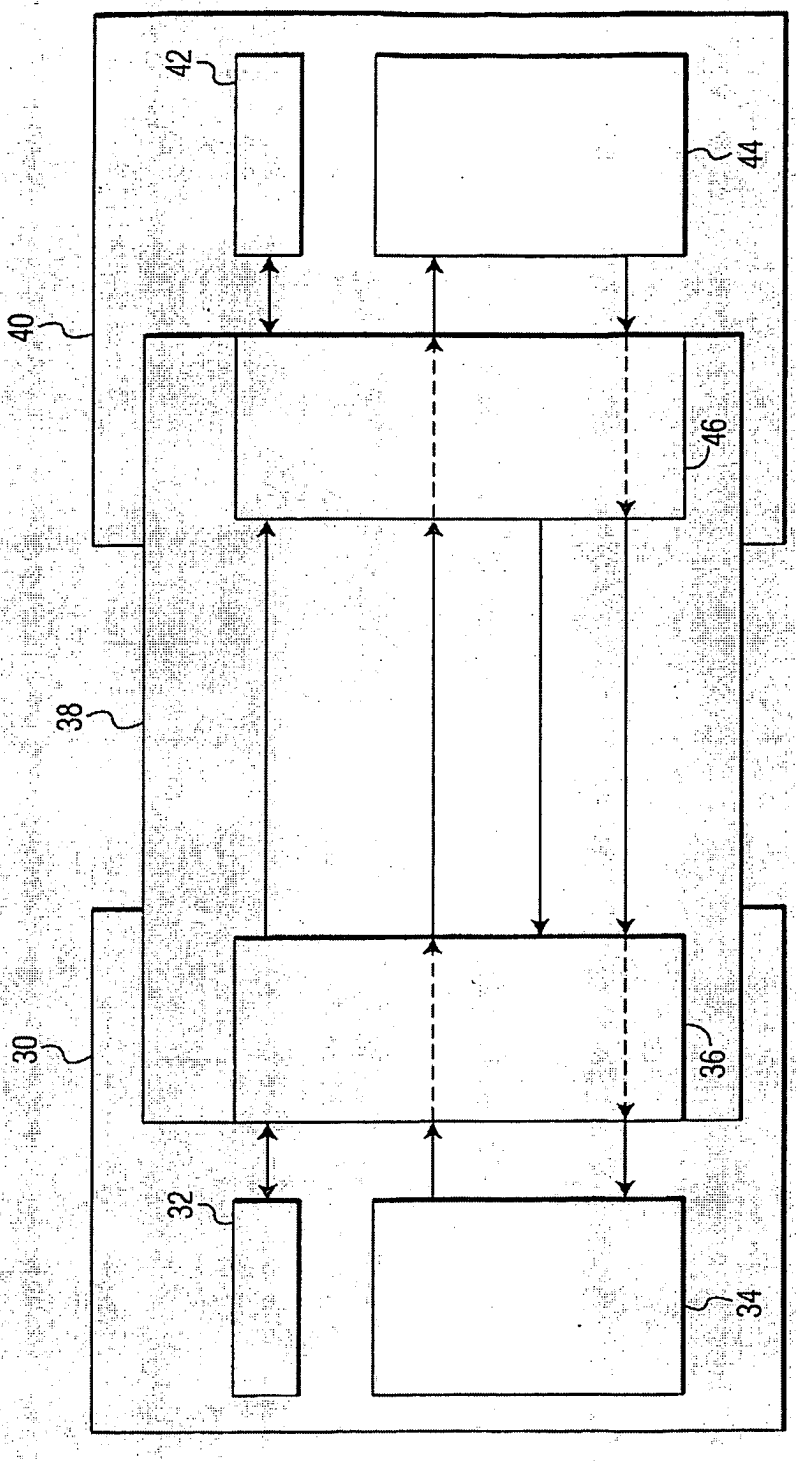


FIG. 3

PCT/IB2004/051819

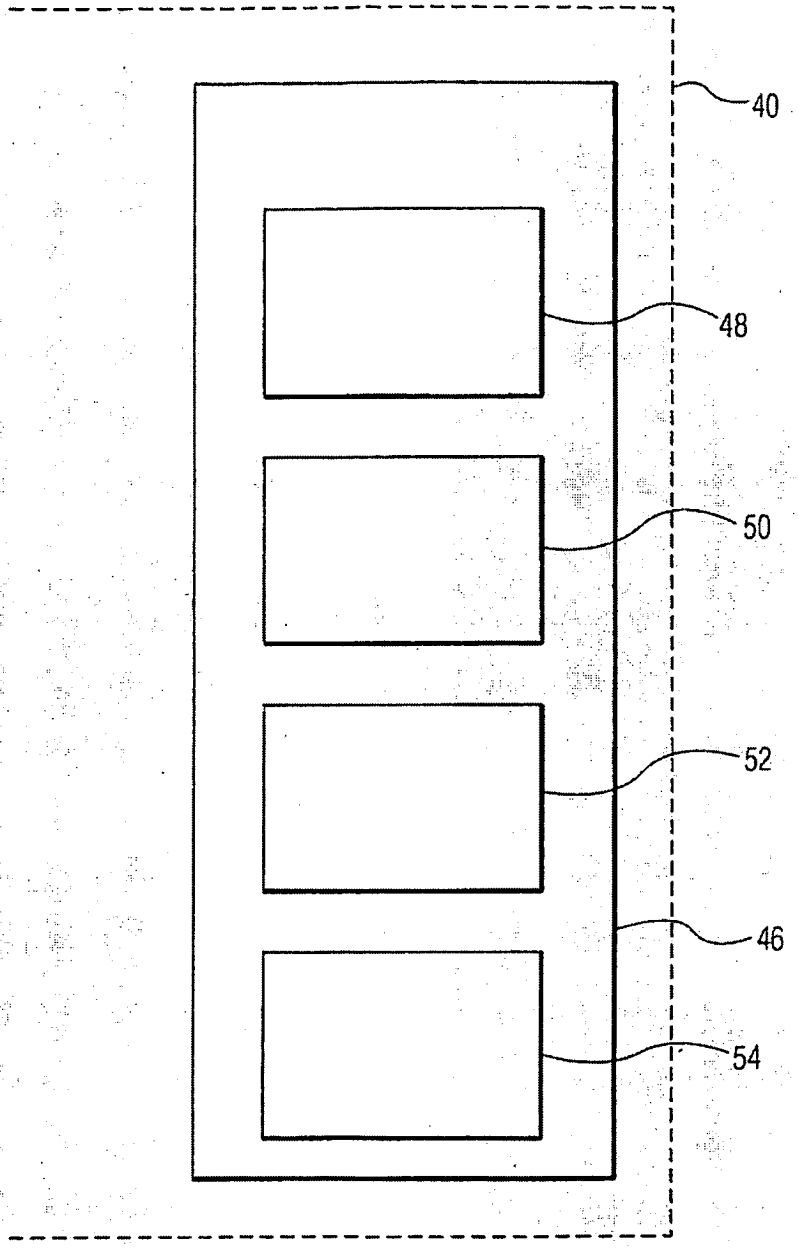


FIG. 4

PCT/IB2004/051819

0X00		0X00		
0X04		0X04		
0X08		0X08		
0X0C		0X0C	–	
0X10		0X10	–	
0X14		0X14		
0X18		0X18		
0X1C		0X1C		
0X20		0X20		
0X24		0X24		
0X28		0X28		
0X2C		0X2C		
0X3D		0X30		
0X34		0X34		
0X38		0X38		
0X3C		0X3C		
0X40		0X40		
0X44		0X44		
0X48		0X48		
0X4C		0X4C		

FIG. 5A

PCT/IB2004/051819

0X50		0X50		
0X54		0X54		
0X58		0X58		
0X5C		0X5C		
0X6D		0X60		
0X64		0X64		
	–	0X68		
		0X6C		
	–	0X70		
		0X74		
	–	0X78		
		0X7C		
	–	0X80		
		0X84		

FIG. 5B

PCT/IB2004/051819

31:10		—	—	0x0*	
9:5	INT_ENABLE			0x0*	
4:0	INT_STATUS			0x0*	

FIG. 6A

31:10		—	—	0x0*	
9:5	INT_ENABLE			0x0*	
4:0	INT_STATUS			0x0*	

FIG. 6B

PCT/IB2004/051819

31:10		—	—	0x0*	
9:8				0x0*	
7	RX_DMA_1 _STATUS			0x0*	
6	RX_DMA_0 _STATUS			0x0*	
5	TX_DMA_1 _STATUS			0x0*	
4	TX_DMA_0 _STATUS			0x0*	
1	NEXT_TX_DMA			0x0*	

FIG. 6C

PCT/IB2004/051819

0	NEXT_RX_DMA			0x0*	

FIG. 6C1

PCT/IB2004/051819

31:10		—	—	0x0*	
9:0	MLBX_CODE			0x0*	

FIG. 6D

31:10		—	—	0x0*	
9:0	SCRCH_CODE			0x0*	

FIG. 6E

31:10		—	—	0x0*	
9:0	MLBX_CODE			0x0*	

FIG. 6F

31:10		—	—	0x0*	
9:0	SCRCH_CODE			0x0*	

FIG. 6G

31:10		—	—	0x0*	
9				0x0*	
8	DMA_msb_first			0x0*	
7	RX_DMA_1 _INT_ENABLE			0x0*	
6	RX_DMA_0 _INT_ENABLE			0x0*	
5	TX_DMA_1 _INT_ENABLE			0x0*	
4	TX_DMA_0 _INT_ENABLE			0x0*	
3	RX_DMA_1 _INT_STATUS			0x0*	
2	RX_DMA_0 _INT_STATUS			0x0*	
1	TX_DMA_1 _INT_STATUS			0x0*	
0	TX_DMA_0 _INT_STATUS			0x0*	

FIG. 6H

PCT/IB2004/051819

31:26		—		0x0*	
25:0	DMA_ADDR	—		0x0*	

FIG. 6I

31:26				0x0*	
25:0	DMA_SIZE			0x0*	

FIG. 6J

31:26		—		0x0*	
25:0	DMA_ADDR	—		0x0*	

FIG. 6K

31:16		—		0x0*	
15:0	DMA_SIZE	—		0x0*	

FIG. 6L

PCT/IB2004/051819

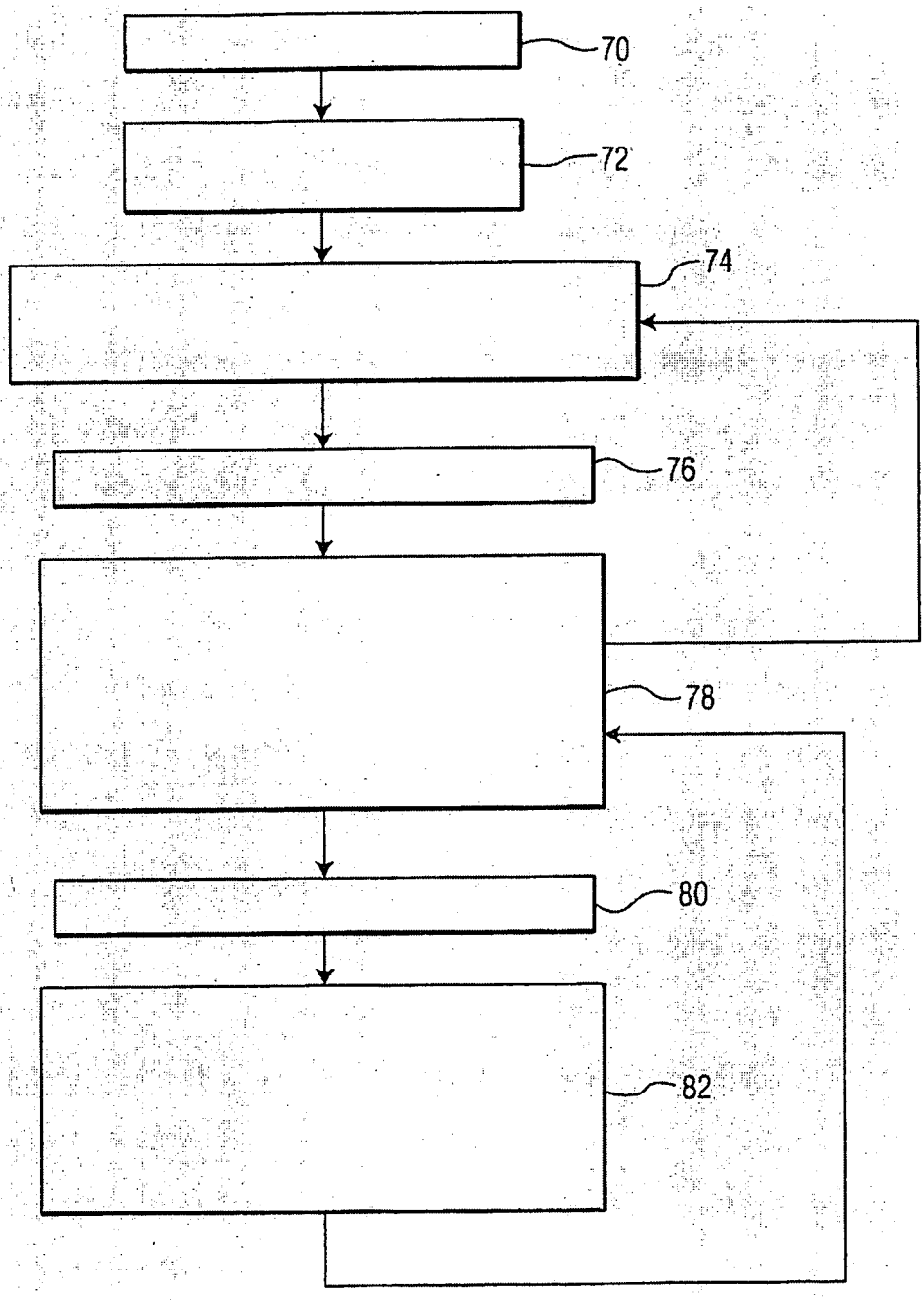


FIG. 7

PCT/IB2004/051819

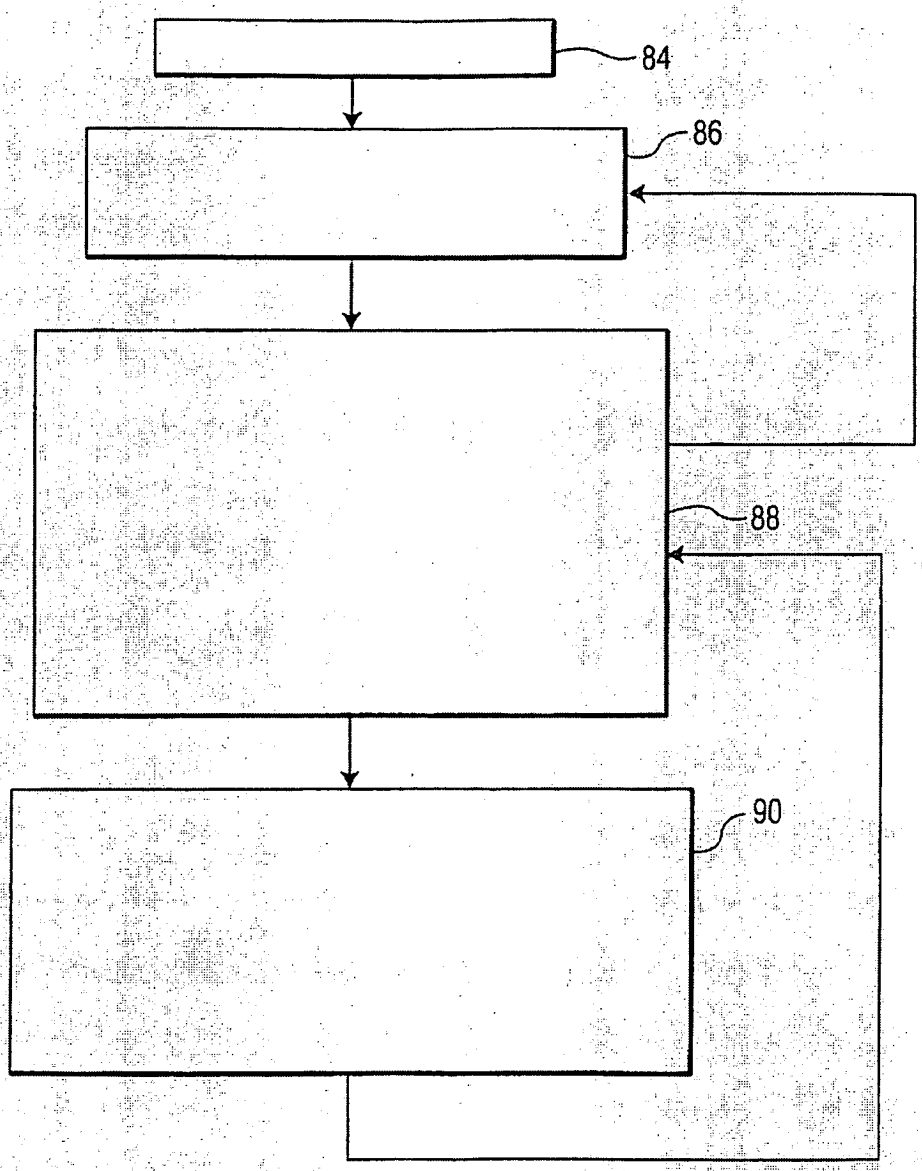


FIG. 8

PCT/IB2004/051819

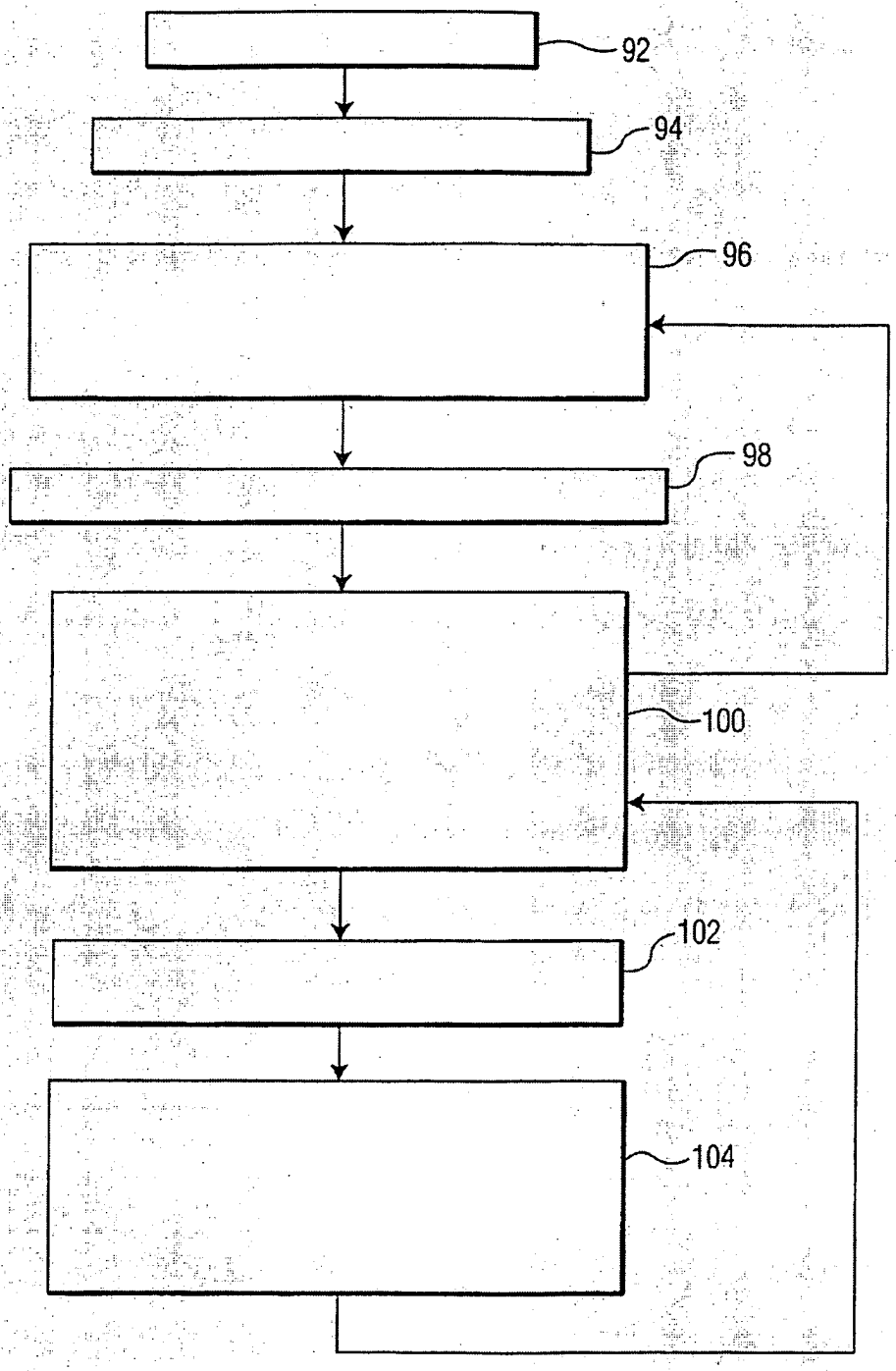


FIG. 9

PCT/IB2004/051819

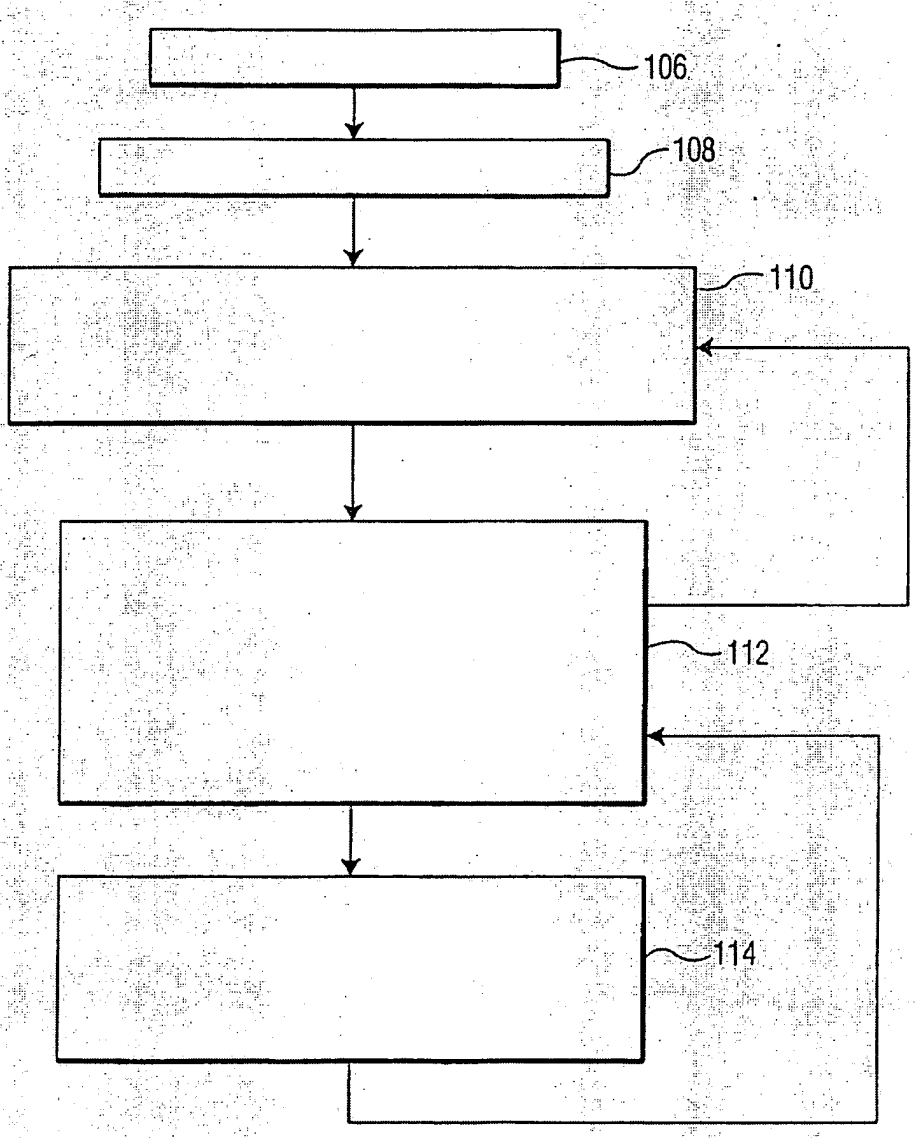


FIG. 10

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F13/14

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
A	US 5 111 425 A (TAKEUCHI ET AL) 5 May 1992 (1992-05-05) abstract column 1, line 8 - column 2, line 19	1-25
A	US 4 782 439 A (BORKAR ET AL) 1 November 1988 (1988-11-01) the whole document	1-25
A	US 4 716 523 A (BURRUS, JR. ET AL) 29 December 1987 (1987-12-29) abstract column 4, line 3 - column 5, line 48	1-25
A	US 2003/093492 A1 (BADE PAUL R ET AL) 15 May 2003 (2003-05-15) the whole document	1-25

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
9 March 2005	16/03/2005

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B 5818 Patentlaan 2 NL -2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx 31 651 epo nl, Fax: (+31-70) 340-3016	Rudolph, S

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

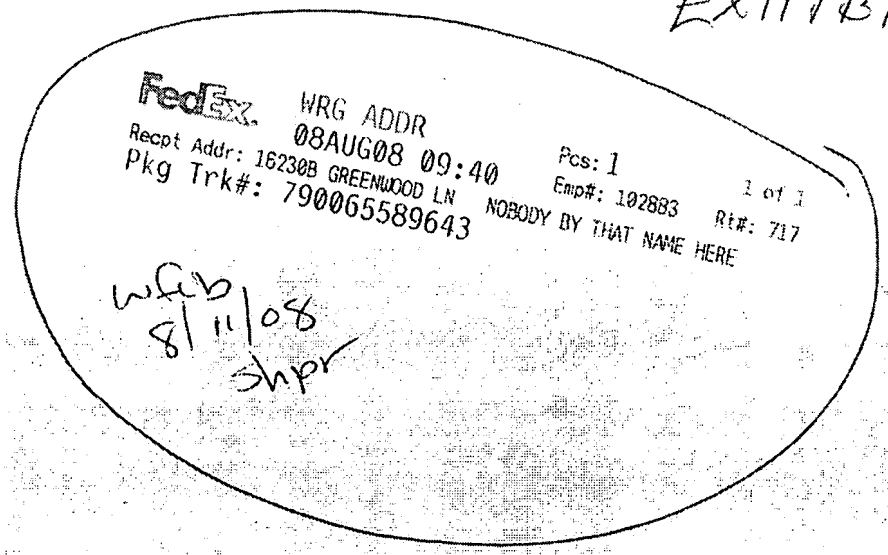
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
A	WO 94/09436 A (COMPAQ COMPUTER CORPORATION) 28 April 1994 (1994-04-28) abstract page 2, line 17 - page 5, line 41 page 9, line 23 - page 11, line 20	1-25

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5111425 A	05-05-1992	JP 2194460 A	01-08-1990
		JP 2712131 B2	10-02-1998
US 4782439 A	01-11-1988	GB 2201269 A ,B	24-08-1988
		JP 63288353 A	25-11-1988
US 4716523 A	29-12-1987	BR 8602560 A	03-02-1987
		CA 1243125 A1	11-10-1988
		DE 3688810 D1	09-09-1993
		DE 3688810 T2	17-03-1994
		EP 0204960 A2	17-12-1986
		JP 1754451 C	23-04-1993
		JP 4050623 B	14-08-1992
		JP 61288248 A	18-12-1986
US 2003093492 A1	15-05-2003	US 2005005036 A1	06-01-2005
		AU 2603901 A	09-07-2001
		WO 0148620 A1	05-07-2001
WO 9409436 A	28-04-1994	AU 5442494 A	09-05-1994
		WO 9409436 A1	28-04-1994
		US 5469548 A	21-11-1995

EXHIBIT 2

EXHIBIT 2

Align top of FedEx Express Shipping



FedEx Express

From: Origin ID: RHVA (408)474-9067
vilimaina Naga
NXP Semiconductors
1109 McKay Drive M/S-41SJ
San Jose,, CA 95131

FedEx Express

Ship Date: 07AUG08
ActWgt: 1 LB
System#: 124494/INET8061
Account#: S *********

Delivery Address Bar Code

SHIP TO: 4084749060 BILL SENDER
Guido Frederiks
16230B GREENWOOD LN
MONTE SERENO, CA 950303053

Ref # US030356US, US040131US
Invoice #
PO #
Dept # 9081

TRK# 0201 7900 6558 9643

FRI - 08AUG
PRIORITY C
RES

WA SJCA